Hub (SOLUSI)

Solusi ini membutuhkan:

- Wireshark
- Python

Langkah:

1. Membaca file .pcapng

Hal yang dilakukan pertama adalah membaca file .pcapng tersebut. Dilihat sekilas, seluruh protokol yang berada dalam file .pcapng ini adalah USB. Terdapat 13 ribu entry packet yang lalu-lalang pada protokol USB, jadi perlu sedikit filtering terhadap data yang ditunjukkan. Cara yang paling mudah adalah melakukan sort terhadap sumber packet, lalu melihat obrolan pada masing-masing USB interface.

о.	Time	Source	estination	Protocol	Length Info
	32 0.000000	1.1.0	lost	USB	46 GET DESCRIPTOR Response DEVICE
	34 0.000000	1.1.0	host	USB	59 GET DESCRIPTOR Response CONFIGURATION
	36 0.000000	1.1.0	host	USB	28 SET CONFIGURATION Response
	2 0.000000	1.2.0	host	USB	46 GET DESCRIPTOR Response DEVICE
	4 0.000000	1.2.0	host	USB	53 GET DESCRIPTOR Response CONFIGURATION
	6 0.000000	1.2.0	host	USB	28 SET CONFIGURATION Response
	20 0.000000	1.3.0	host	USB	46 GET DESCRIPTOR Response DEVICE
	22 0.000000	1.3.0	host	USB	87 GET DESCRIPTOR Response CONFIGURATION
	24 0.000000	1.3.0	host	USB	28 SET CONFIGURATION Response
	1953 6.231754	1.3.1	host	USB	35 URB_INTERRUPT in
	1973 6.303635	1.3.1	host	USB	35 URB_INTERRUPT in
	2183 7.119747	1.3.1	host	USB	35 URB_INTERRUPT in
	2203 7.191638	1.3.1	host	USB	35 URB_INTERRUPT in
	7839 22.559757	1.3.1	host	USB	35 URB_INTERRUPT in
	7859 22.631765	1.3.1	host	USB	35 URB_INTERRUPT in
	7877 22 670653	1 2 1	host	HCR	RE HIDE THITEDDINGT in

2. Mengekstrak hal yang menarik

Setelah melakukan skimming terhadap PCAP yang telah disort, terdapat sebuah sumber dengan alamat 1.6.1 yang setelah dibaca DESCRIPTOR nya adalah sebuah USB drive. Apa yang biasanya dilakukan USB drive? Yak, transfer data. Dengan mengaplikasikan filter usb.addr == 1.6.1, kita bisa melihat apa yang dilakukan USB ini pada komputernya.

```
usb.addr == "1.6.1"
                                                     Destination
                                                                             Protocol
                                                                                                   Length Info
        5408 15.464959
        5409 15.464987
                              host
                                                     1.6.1
                                                                             USB
                                                                                                       27 URB BULK in
        5410 15.465107
                             1.6.1
                                                     host
                                                                             USBMS
                                                                                                       40 SCSI: Response LUN: 0x00 (Write(10)) (Good)
        5411 15.465298
5412 15.465364
                             host
1.6.1
                                                      1.6.1
                                                                                                       58 SCSI: Write(10) LUN: 0x00 (LBA: 0x00
27 URB_BULK out
                                                                             USBMS
        5413 15.465366
                             host
                                                     1.6.1
                                                                                                    4123 SCSI: Data Out LUN: 0x00 (Write(10) Request Data)
                                                                                                       27 URB_BULK out
27 URB_BULK in
         5414 15 465475
                             1 6 1
                                                                              IISR
                                                     1.6.1
        5416 15.466607
                             1.6.1
                                                     host
                                                                             USBMS
                                                                                                       40 SCSI: Response LUN: 0x00 (Write(10)) (Good)
        5417 15.467439
                             host
                                                     1.6.1
                                                                             USBMS
                                                                                                       58 SCSI: Write(10) LUN: 0x00 (LBA: 0x00004000, Len: 8)
        5418 15.467499
5419 15.467501
                                                     host
1.6.1
                                                                             USB
USBMS
                                                                                                    27 URB_BULK out
4123 SCSI: Data Out LUN: 0x00 (Write(10) Request Data)
        5420 15.467637
                             1.6.1
                                                     host
                                                                             USB
                                                                                                       27 URB BULK out
                                                     1.6.1
host
                                                                                                       27 URB_BULK in
40 SCSI: Response LUN: 0x00 (Write(10)) (Good)
         5421 15.467638
                             host
1.6.1
        5423 15.467775
                             host
                                                     1.6.1
                                                                             USBMS
                                                                                                       58 SCSI: Write(10) LUN: 0x00 (LBA: 0x000043c0, Len: 1)
        5424 15.467795
                             1.6.1
                                                     host
                                                                             USB
                                                                                                       27 URB BULK out
        5425 15.467796
                                                                                                      539 SCSI: Data Out LUN: 0x00 (Write(10) Request Data)
         5426 15.467852
                             1.6.1
                                                                                                       27 URB_BULK out
        5427 15.467853
                             host
                                                     1.6.1
                                                                             USB
                                                                                                       27 URB BULK in
        5428 15.467901
                                                                                                       40 SCSI: Response LUN: 0x00 (Write(10)) (Good)
  Frame 5425: 539 bytes on wire (4312 bits), 539 bytes captured (4312 bits) on interface wireshark_extcap3588, id 0 USB URB
  USB Mass Storage
```

Dengan sedikit melihat-lihat, terdapat sebuah packet yang berisikan sebuah zipfile yang bisa diekstraksi. Ekstraksi yang paling jelas adalah dengan mengopi hex dari data tersebut kemudian ditaruh ke dalam sebuah file.

```
ziphex =
```

```
zipdata = bytes.fromhex(ziphex)
f = open("out.zip", "wb")
f.write(zipdata)
f.close()
```

> SCSI Payload (Write(10) Request Data)

- 3. Mengetahui bahwa file yang dikirimkan adalah zip yang dikunci dengan sebuah password Setelah dicoba dibuka, ternyata zip ini berpassword dan dienkripsi dengan AES-256. Dengan cara naif, mungkin beberapa akan mencoba untuk melakukan John the Ripper atau fcrackzip pada file ini. Namun karena passwordnya tidak berada di dalam dictionary manapun, mungkin peserta bisa terjebak pada rabbithole ini.
- 4. Mengekstrak dan menerjemahkan ketikan keyboard yang ditangkap pada file .pcapng tersebut

Melihat hint yang diberikan, atau mungkin mendapat pencerahan dari langit, selain USB drive, terdapat juga USB device lain yang ditangkap pada file PCAP ini, salah satunya keyboard. Device lain yang terdapat pada file ini adalah mouse dan gamepad. Dengan menerapkan filter usb.src == "1.3.1", kita bisa melihat bahwa terdapat data yang dikirimkan oleh keyboard kepada host, yang menunjukkan bahwa seseorang sedang mengetik sesuatu pada saat USB di-capture.

Time	Source	Destination	Protocol	Length Info	
1953 6.2317	54 1.3.1	host	USB	35 URB_INTERRUPT	in
1973 6.3036	35 1.3.1	host	USB	35 URB_INTERRUPT	in
2183 7.1197	47 1.3.1	host	USB	35 URB_INTERRUPT	in
2203 7.1916	38 1.3.1	host	USB	35 URB_INTERRUPT	in
7839 22.559	757 1.3.1	host	USB	35 URB_INTERRUPT	in
7859 22.631	765 1.3.1	host	USB	35 URB_INTERRUPT	in
7877 22.679	653 1.3.1	host	USB	35 URB_INTERRUPT	in
7901 22.767	652 1.3.1	host	USB	35 URB_INTERRUPT	in
7933 22.887	763 1.3.1	host	USB	35 URB_INTERRUPT	in
7953 22.959	765 1.3.1	host	USB	35 URB_INTERRUPT	in
7963 22.991	676 1.3.1	host	USB	35 URB_INTERRUPT	in
7989 23.087	648 1.3.1	host	USB	35 URB_INTERRUPT	in
7995 23.103	663 1.3.1	host	USB	35 URB_INTERRUPT	in
8033 23.239	834 1.3.1	host	USB	35 URB_INTERRUPT	in
8043 23.279	652 1.3.1	host	USB	35 URB_INTERRUPT	in
8065 23.359	770 1.3.1	host	USB	35 URB_INTERRUPT	in
8421 24.743	759 1.3.1	host	USB	35 URB_INTERRUPT	in
8443 24.823	765 1.3.1	host	USB	35 URB_INTERRUPT	in
8455 24.863	648 1.3.1	host	USB	35 URB_INTERRUPT	in
8479 24.951	698 1.3.1	host	USB	35 URB_INTERRUPT	in
8501 25.031	684 1.3.1	host	USB	35 URB_INTERRUPT	in
				_	
Enamo 7030. 3E	urtes en urine /200 hi	ts\ 25 butos contunad	/200 bits) on inton	Face wineshank outcanace	: 4 0
rrame 7639: 35 I USB URB	yces on wire (200 bi	cs), 33 bytes captured	(200 DIES) ON INTER	face wireshark_extcap3588,	IU 0
HID Data: 00000	lanananana				

Keyboard tidak mengirimkan data secara gamblang, namun berdasarkan standar USB keyboard. (https://www.usb.org/sites/default/files/documents/hut1_12v2.pdf) (Halaman 53)

			Ref: Typical AT-101				
Usage ID (Dec)	Usage ID (Hex)	Usage Name	Position	PC-	Мас	UN X	Boot
0	- 00	Description over indicated)9	N/A	√.	V		4/101/104
		Reserved (no event indicated) ⁹					
1	01	Keyboard ErrorRollOver ⁹	N/A	√.	1		4/101/104
2	02	Keyboard POSTFail ⁹	N/A	V	V	1	4/101/104
3	03	Keyboard ErrorUndefined9	N/A	V	V	V	4/101/104
4	04	Keyboard a and A ⁴	31	V	V	V	4/101/104
5	05	Keyboard b and B	50	\checkmark	V	V	4/101/104
6	06	Keyboard c and C ⁴	48	V	V		4/101/104
7	07	Keyboard d and D	33	\checkmark	V	V	4/101/104
8	80	Keyboard e and E	19	V	V	V	4/101/104
9	09	Keyboard f and F	34	\checkmark	V		4/101/104
10	0A	Keyboard g and G	35	V	V		4/101/104
11	0B	Keyboard h and H	36	V	V		4/101/104
12	0C	Keyboard i and I	24	\checkmark	V	V	4/101/104
13	0D	Keyboard j and J	37	\checkmark	V		4/101/104
14	0E	Keyboard k and K	38	\checkmark	V	V	4/101/104
15	0F	Keyboard I and L	39	\checkmark	V		4/101/104
16	10	Keyboard m and M ⁴	52	\checkmark	V		4/101/104
17	11	Keyboard n and N	51	\checkmark	V		4/101/104
18	12	Keyboard o and O ⁴	25	√	V		4/101/104
19	13	Keyboard p and P4	26	√	V		4/101/104
20	14	Keyboard q and Q ⁴	17	V	V		4/101/104

Kemudian data yang telah disaring dapat diexport menjadi JSON (File > Export Packet Dissection > As JSON...), kemudian diterjemahkan dengan script sebagai berikut:

```
import json

switcher = {'04': 'a', '05': 'b', '06': 'c', '07': 'd', '08': 'e', '09': 'f', '0A': 'g',
    '08': 'h', '0C': 'i', '0D': 'j', '0E': 'k', '0F': 'l', '10': 'm', '11': 'n', '12': 'o', '13':
    'p', '14': 'q', '15': 'r', '16': 's', '17': 't', '18': 'u', '19': 'v', '1A': 'w', '1B': 'x',
    '1C': 'y', '1D': 'x', '1E': '1', '1F': '2', '20': '3', '21': '4', '22': '5', '23': '6', '24':
    '7', '25': '8', '26': '9', '27': '0', '2D': '-', '2E': '+', '2F': '[', '30': ']', '31': '"',
    '33': ';', '34': "'", '35': '`', '36': ',', '37': '.', '38': '/'}

f = open("keyboard.json")

data = json.loads(f.read())

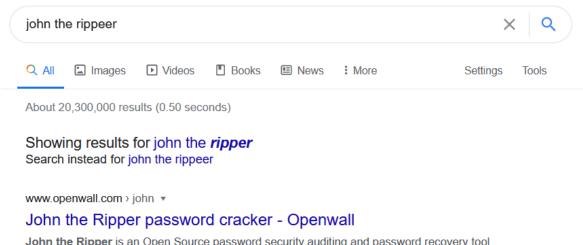
passwd = ""

for i in data:
    try:
        passwd += switcher[i["_source"]["layers"]["usbhid.data"][6:8].upper()]

    except:
        pass

print(passwd)
```

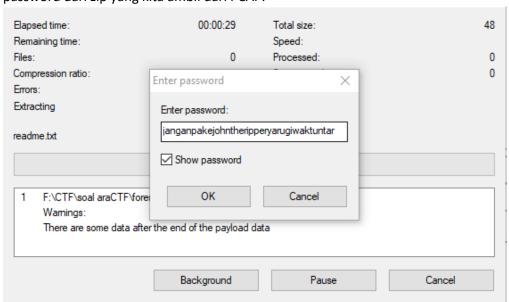
Output dari script tersebut adalah "janganpakejohntherippeeryarugiwaktuntar". Karena sifat alamiah keyboard yang terkadang mengirimkan packet yang sama kedua kali ketika sebuah key ditekan terlalu lama, terkadang ada huruf yang muncul dobel pada packet. John the Ripper tidak dieja dengan dua huruf e, dapat digoogle ejaannya. Kata-kata yang seharusnya muncul pada keyboard adalah "janganpakejohntheripperyarugiwaktuntar".



John the Ripper is an Open Source password security auditing and password recovery tool available for many operating systems. **John the Ripper** jumbo ...

Browse the documentation for ... - John the Ripper Pro - John-users mailing list

5. Mengekstrak flag dengan password
Dari kata-kata yang diekstrak tadi, "janganpakejohntheripperyarugiwaktuntar" adalah
password dari zip yang kita ambil dari PCAP.



File readme.txt merupakan flagnya.

FLAG:

ara2021{password_zip_alay_tapi_flag_jangan_alay}